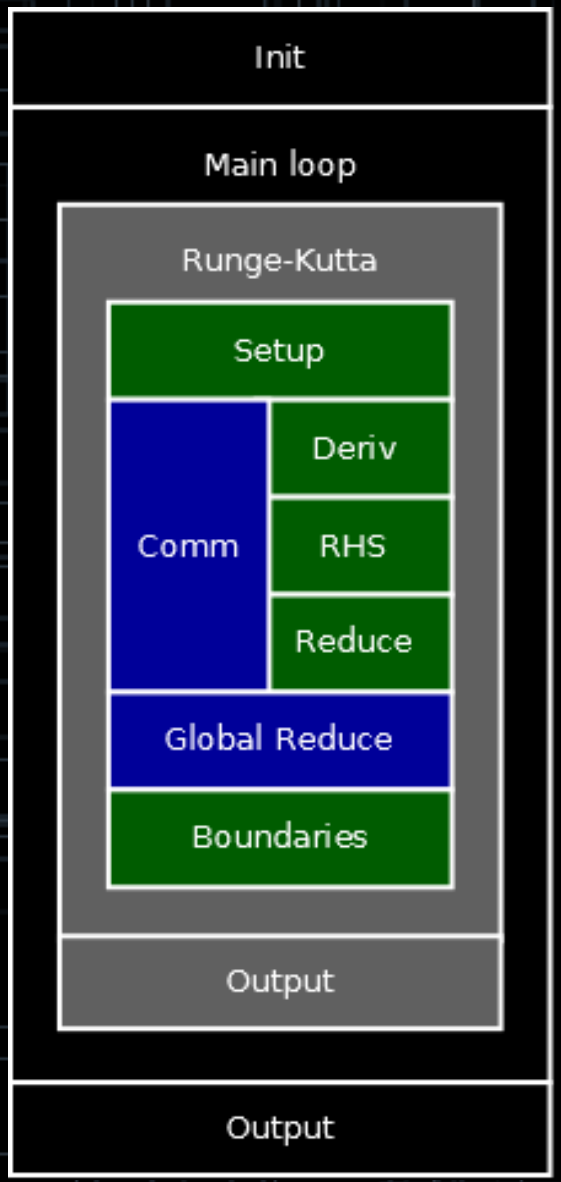
Simulación de la ecuación del flujo de Ricci en S^3 usando GPU

Carlos Bederián

14 – 11 - 2010

Estructura



x 8

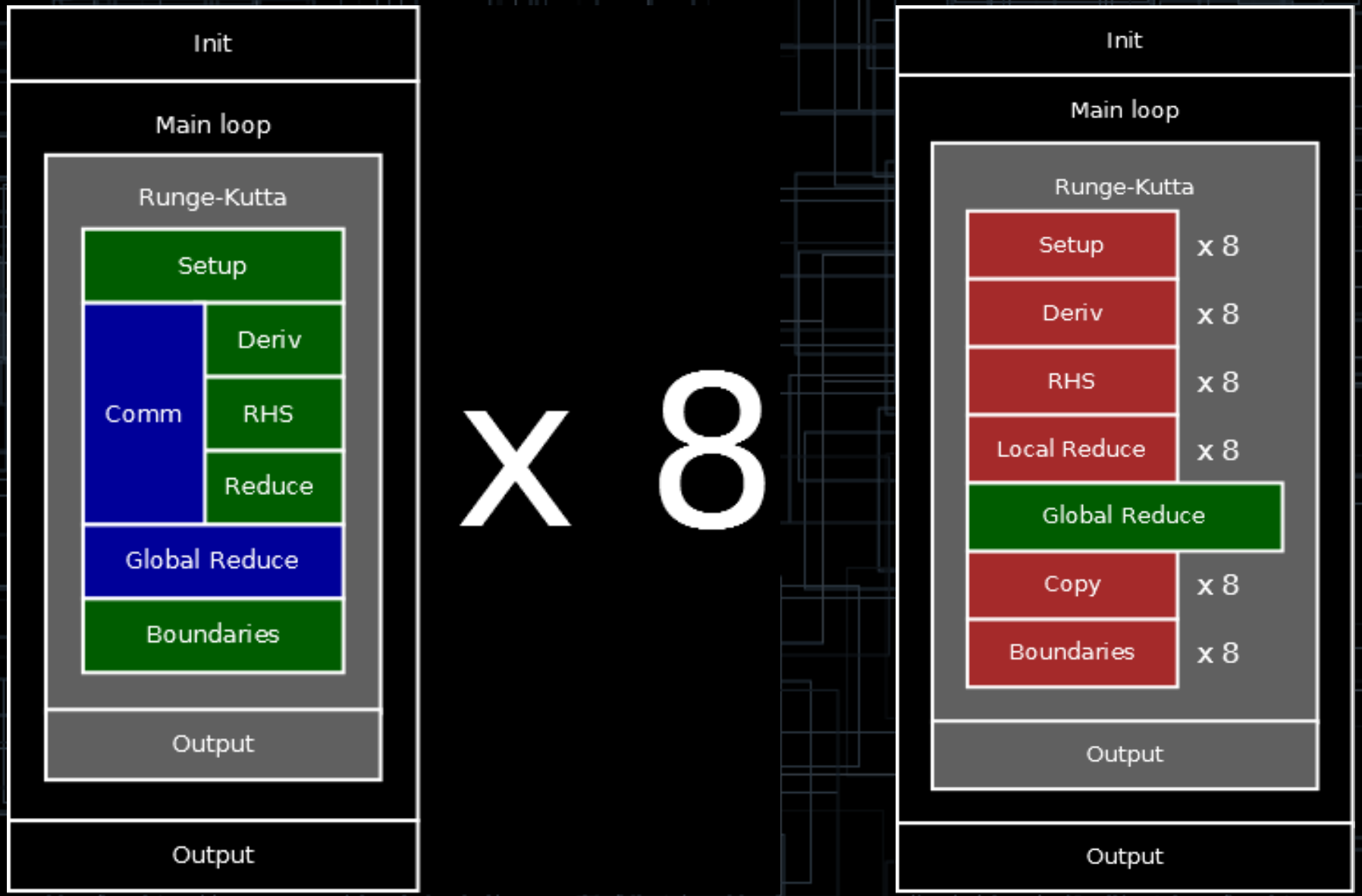
El plan

1. Dividir el problema en funciones
2. Convertir funciones en kernels
3. Debugging
4. Profiling / Mejoras
5. Profit

Lo que pasó

1. Dividir el problema en funciones
2. Convertir funciones en kernels
3. ~~Debugging~~
4. Eliminar MPI
5. Debugging
6. Profiling / Mejoras
7. Agregar MPI
8. Profit

Estructura


Init
Main loop
Runge-Kutta
Setup
Comm
Deriv
RHS
Reduce
Global Reduce
Boundaries
Output
Output
x 8
Init
Main loop
Runge-Kutta
Setup
x 8
Deriv
x 8
RHS
x 8
Local Reduce
x 8
Global Reduce
Copy
x 8
Boundaries
x 8
Output
Output

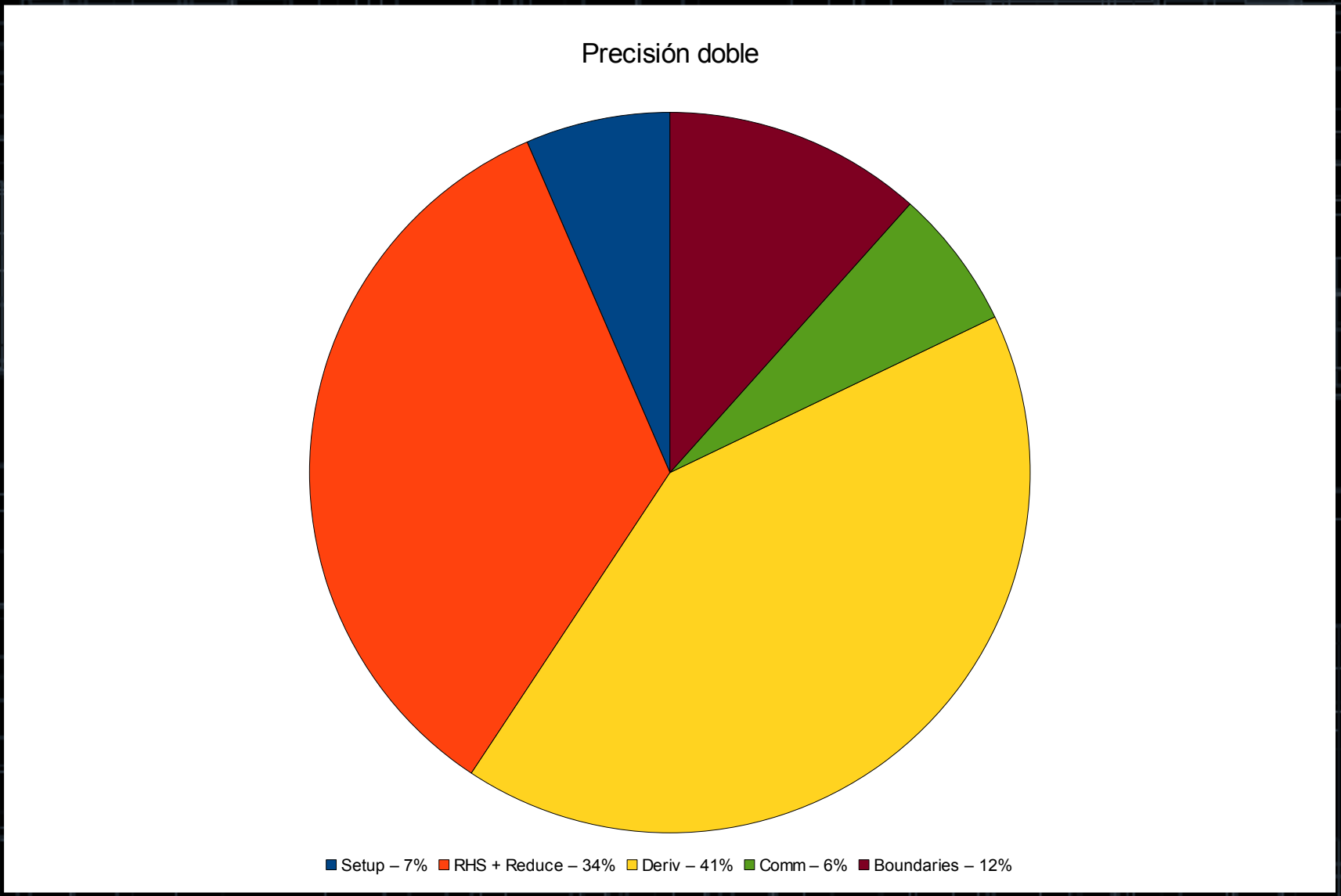
Dificultades

- gridDim.z debe ser 1
  - Kernels con loop sobre z
  - Correr gridDim.y x gridDim.z bloques en y
- Uso excesivo de registros
  - Occupancy baja
  - El compilador hace spilling a local memory!
  - Tratar de hacer el menor trabajo posible en estos kernels

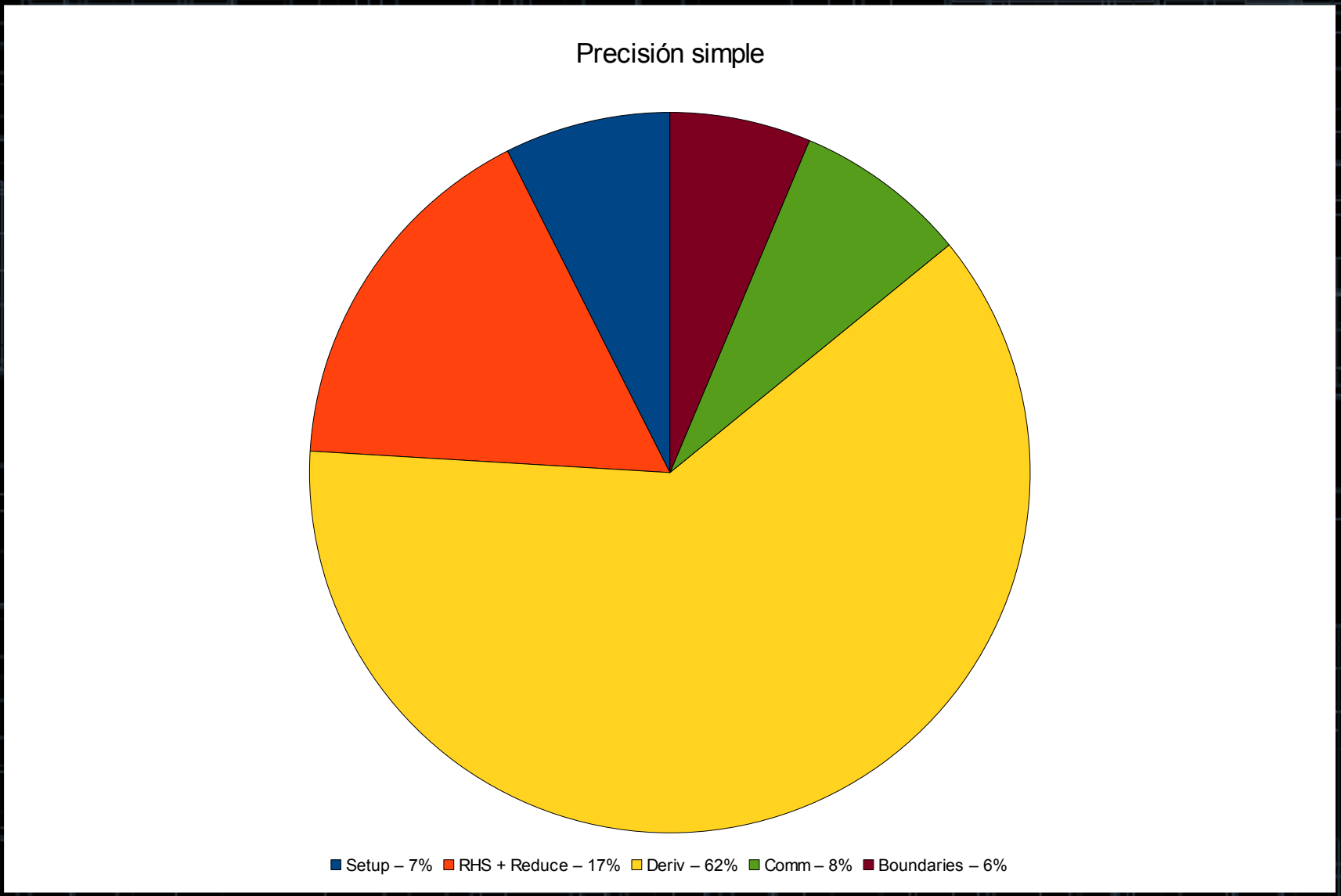
Dificultades

- Uso excesivo de memoria
  - Eliminar datos duplicados
  - Recalcular campos sencillos
  - Acumular donde sea posible
  - → 50% menos memoria!
- Código multiprecisión
  - Los literales de doble precisión eliminan toda la ventaja de usar floats
  - → Regar el código con macros

¿En qué concentrarse?



Profile early, profile often



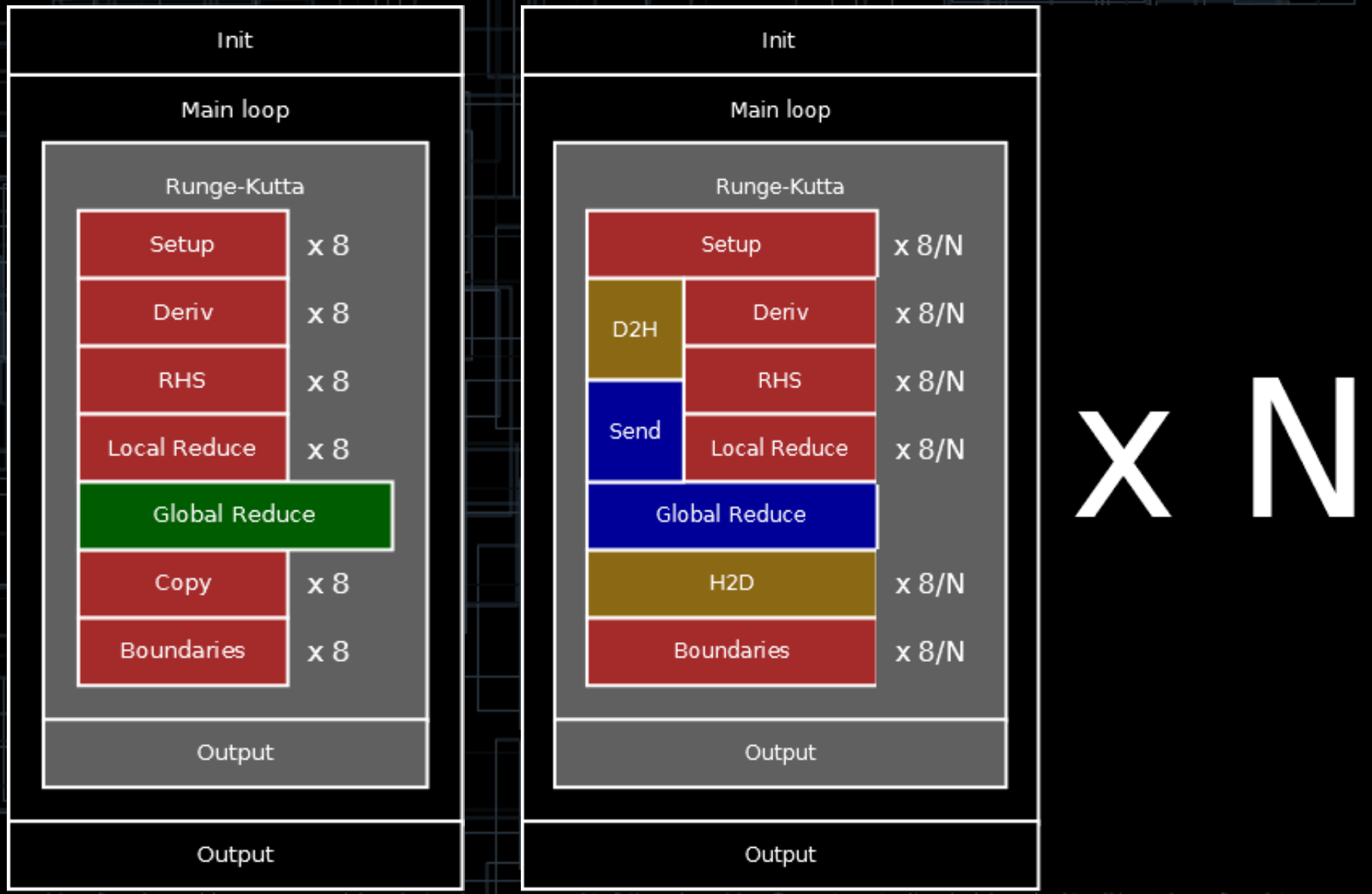
Derivadas

- 8 / 12 datos por punto
  - División entre cálculo de valores internos y externos
- Memory bound
  - Uso de shared memory
  - 25-45 GB/s en algunos casos, se puede mejorar

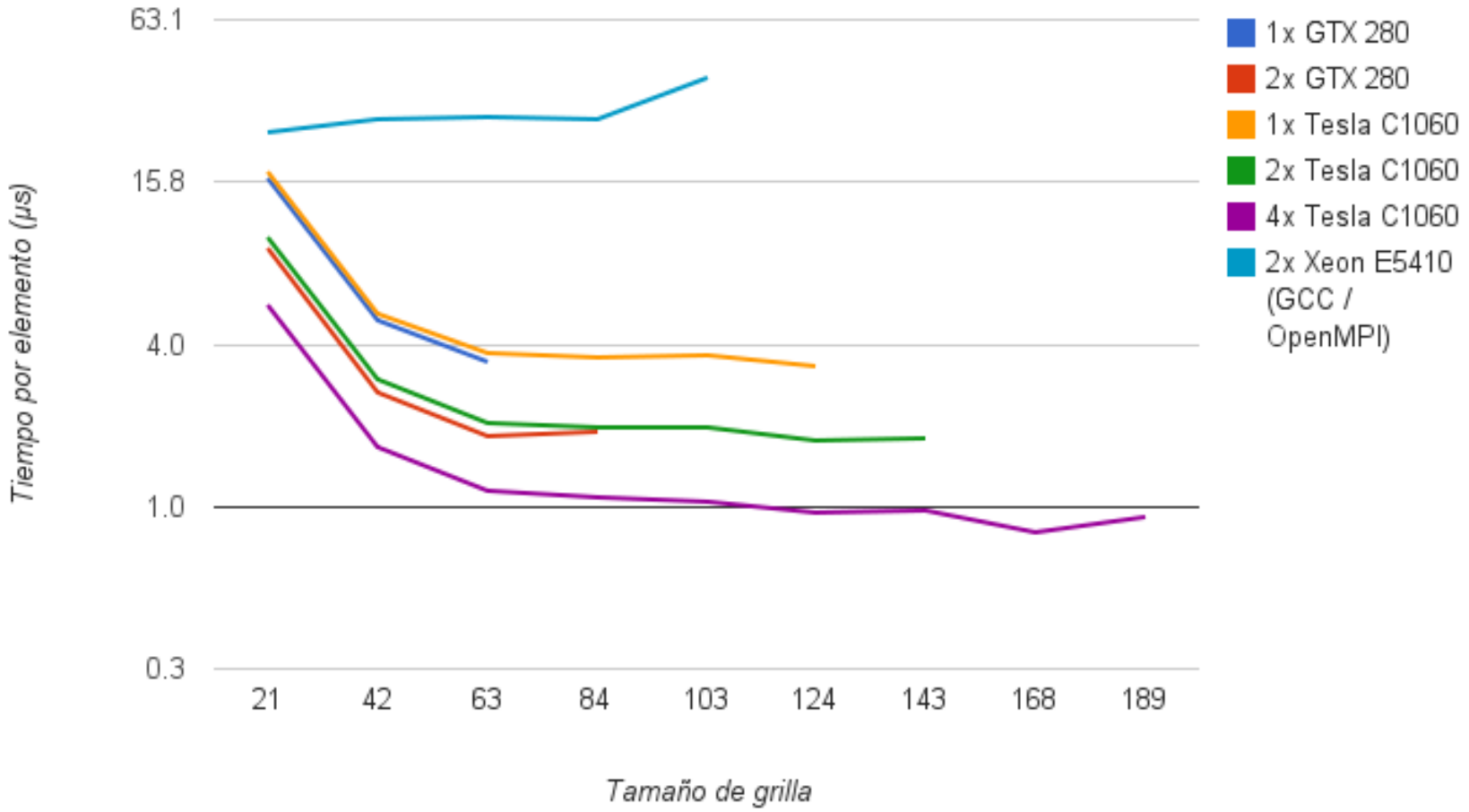
Reinserción de MPI

- Conservar la mayor localidad posible
  - 8 / N grids por proceso
- Código MPI aislado
  - Funciones básicas envueltas
  - Fácil de apagar para debugging
- Paralelismo de computación y comunicación
  - Uso de streams y pinned memory para copias asíncronas al host
  - Nonblocking I/O
- 2 GPUs: >1.88X, 4 GPUs: >3.44X

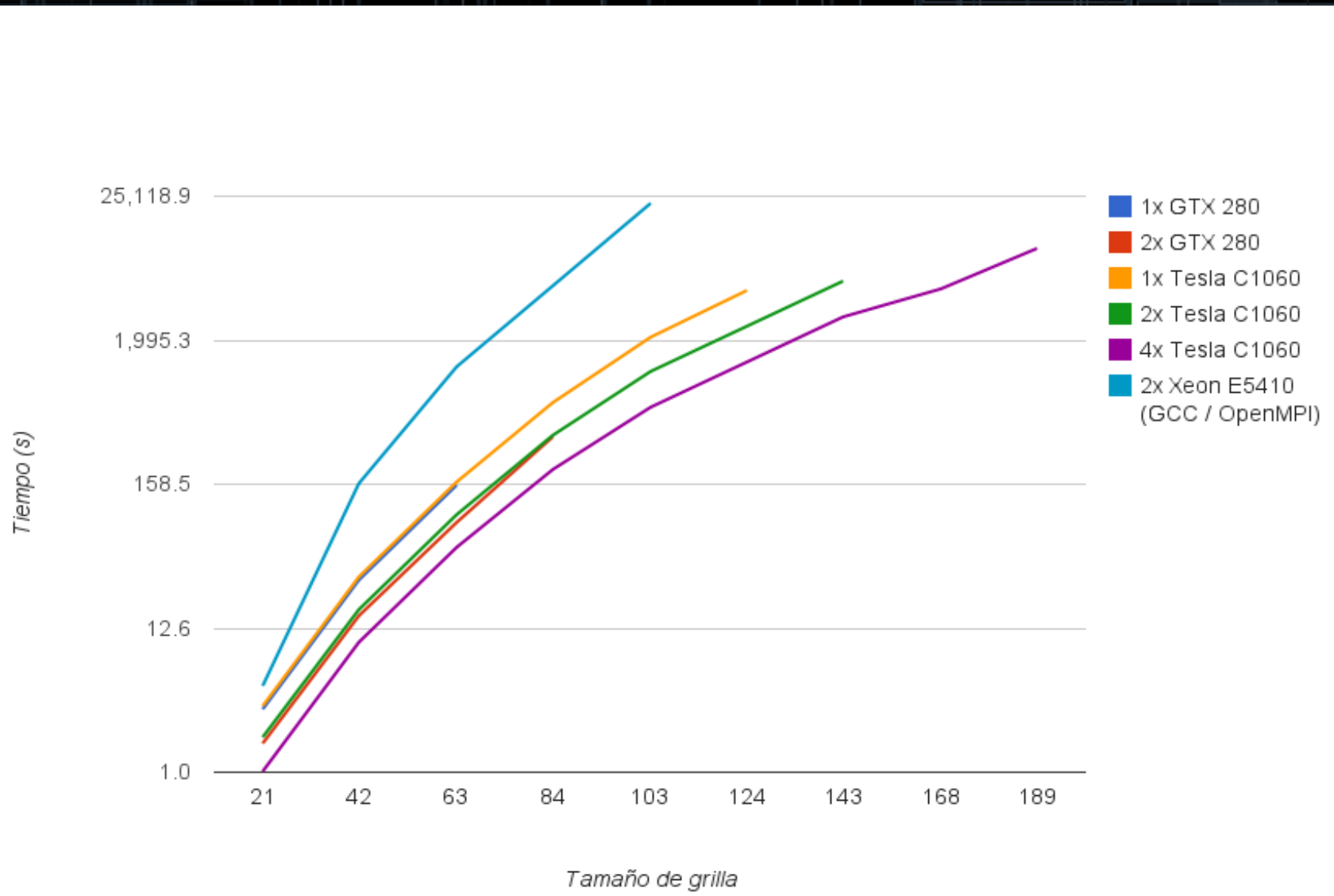
Estructura


Init
Main loop
Runge-Kutta
Setup
x 8
Deriv
x 8
RHS
x 8
Local Reduce
x 8
Global Reduce
Copy
x 8
Boundaries
x 8
Output
Output
Init
Main loop
Runge-Kutta
Setup
x 8/N
D2H
Deriv
x 8/N
RHS
x 8/N
Send
Local Reduce
x 8/N
Global Reduce
H2D
x 8/N
Boundaries
x 8/N
Output
Output
x N

Resultados


63.1
15.8
4.0
1.0
0.3
Tiempo por elemento (µs)
21
42
63
84
103
124
143
168
189
Tamaño de grilla
1x GTX 280
2x GTX 280
1x Tesla C1060
2x Tesla C1060
4x Tesla C1060
2x Xeon E5410 (GCC / OpenMPI)

Resultados


25,118.9
1,995.3
158.5
12.6
1.0
Tiempo (s)
21
42
63
84
103
124
143
168
189
Tamaño de grilla
1x GTX 280
2x GTX 280
1x Tesla C1060
2x Tesla C1060
4x Tesla C1060
2x Xeon E5410
(GCC / OpenMPI)